

Amendment to the Specification

On page 1, please amend the title to read as follows:

METHOD FOR DYNAMIC MANAGEMENT OF TCP REASSEMBLY BUFFERS

On page 1, please amend paragraph [0001] as follows:

[0001] The present invention is related to data transfer. More particularly, the present invention provides a method ~~and system~~ for dynamic management of Transmission Control Protocol (TCP) reassembly buffers in hardware (e.g., in a TCP/IP offload engine (TOE)).

On pages 2-4, please delete paragraphs [0008] and [0009], and amend paragraphs [0005]-[0007] and [0010] as follows:

[0005] The present invention provides a method ~~and system~~ for the dynamic management of TCP reassembly buffers in hardware. Dynamic memory management significantly improves the flexibility and scalability of available memory resources. The major challenge of dynamic memory management of TCP reassembly buffers in hardware, however, is to reduce its associated performance penalty and to bring its performance as close as possible to the performance achieved by static memory management methods. The present invention accomplishes these goals.

[0006] The present invention provides a method ~~and system~~ for flexible dynamic memory management of TCP reassembly buffers, allowing efficient hardware implementation. In addition, the method ~~and system~~ of the present invention allow combined dynamic and static memory management using the same hardware implementation. The decision to use dynamic or

static memory management is done on a per reassembly buffer basis to further increase the efficiency of the hardware implementation.

[0007] A first aspect of the present invention is directed to a method for dynamically managing a reassembly buffer, comprising: providing a plurality of data blocks and an indirect list; pointing, via entries in the indirect list, to allocated data blocks in the plurality of data blocks that currently store incoming data; if a free data block in the plurality of data blocks is required for the storage of incoming data, allocating the free data block for storing incoming data; and, if an allocated data block in the plurality of data blocks is no longer needed for storing incoming data, deallocating the allocated data block such that the deallocated data block becomes a free data block.

~~[0008] A second aspect of the present invention is directed to a system for dynamically managing a reassembly buffer, comprising: a plurality of data blocks; an indirect list having a plurality of entries; and a memory manager for controlling allocation and deallocation of the plurality of data blocks; wherein, if a free data block in the plurality of data blocks is required for the storage of incoming data, the memory manager allocates the free data block for storing incoming data; and wherein, if an allocated data block in the plurality of data blocks is no longer needed for storing incoming data, the memory manager deallocates the allocated data block such that the deallocated data block becomes a free data block.~~

~~[0009] A third aspect of the present invention provides a computer program product for performing the methods of the present invention.~~

[0010] A ~~fourth~~ second aspect of the present invention provides a method for storing out-of-order data segments in a reassembly buffer, comprising: providing a plurality of data blocks and an indirect list having a plurality of entries; providing each data segment with a sequence

number, wherein the sequence number specifies which entry in the indirect list is to be associated with the data segment; determining if any of the plurality of data blocks has already been allocated to the specified entry in the indirect list; if a data block has already been allocated to the specified entry, storing the data segment in the allocated data block; and if a data block has not already been allocated to the specified entry, allocating a free data block for the storage of the data segment, and storing the data segment in the allocated free data block.